





.NET PROGRAMMING SQL PROGRAMMING AND DBMS

SQL SELECT STATEMENTS

Chia Yuen Kwan isscyk@nus.edu.sg

(Total Slides=137) 22/02/2018 1:24 pm - c:\users\isscyk\desktop\new sqlp\03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.







- Introduction
- Basic Queries
- Basic Queries (Illustrations with Dafesty Video Rental Shop)
- Advanced Queries
- Advanced Queries (Illustrations with Dafesty Video Rental Shop)





- Upon completion of this lesson, students should be able to use SQL for querying database Tables:
 - Selectively via columns and rows
 - Using Alias on columns and Tables
 - Using Joins
 - INNER JOINS, LEFT OUTER JOIN, RIGHT OUTER JOINS
 - FULL JOINS and CROSS JOINS
 - · Involving Functions such as
 - Aggregate Functions
 - Mathematical, String and system Functions
 - Using Sub Queries
 - having conditions such as AND, OR, IN and BETWEEN conditions

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

3





- Use of the SELECT statement:
 - Retrieves data from one or more rows. Every SELECT statement produces a table of query results containing one or more columns and zero or more rows.
 - Note that the commands are NOT CASE SENSITIVE

SELECT Command

```
SELECT {[ALL, DISTINCT]} [(select-column,...) /*]
FROM (table,....)
{WHERE (search condition) {[AND|OR] (search condition)...}
{GROUP BY (group-column,....)}
{HAVING (search condition)}
{ORDER BY (sort-column,....)}
```





- Introduction
- Basic Queries
- Basic Queries (Illustrations with Dafesty Video Rental Shop)
- Advanced Queries
- Advanced Queries (Illustrations with Dafesty Video Rental Shop)

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

5





SELECT * FROM Customers

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Select all rows and columns in a table

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24





SELECT Id, Name FROM Customers

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Select particular columns of all rows in a table

Result

Id	Name
DN001	John Chia
DN002	Tom
DN003	Jane

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

7





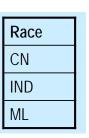


SELECT DISTINCT Race FROM Customers

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Tom	CN
DN003	Jane	IND
DN004	Shawn	ML

Select distinct values in a table







SELECT * FROM Customers WHERE Race = 'CN'

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Tom	CN
DN003	Jane	IND
DN004	Shawn	ML

To conditionally select data in a table

Result

ld	Name	Race
DN001	John Chia	CN
DN002	Tom	CN

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

9





 SELECT * FROM Customers WHERE NOT (Race = 'CN')

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Tom	CN
DN003	Jane	IND
DN004	Shawn	ML

NOT condition can be specified

ld	Name	Race
DN003	Jane	IND
DN004	Shawn	ML





 SELECT Id AS CustomerID, Name FROM Customers

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Alias redefines the result set column name

Alias

Result

CustomerID	Name	
DN001	John Chia	Column name
DN002	Tom	has been changed
DN003	Jane	

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

11





 SELECT * FROM Customers WHERE Race = 'CN' AND Age >=21

The database table "Customers"

ld	Name	Age	Race
DN001	John Chia	21	CN
DN002	Tom	18	CN
DN003	Jane	24	IND
DN004	Shawn	20	ML

AND operator allowed.

ld	Name	Age	Race
DN001	John Chia	21	CN





 SELECT * FROM Customers WHERE Race ='CN' OR Age >=21
The database table "Customers"

Id	Name	Age	Race
DN001	John Chia	21	CN
DN002	Tom	18	CN
DN003	Jane	24	IND
DN004	Shawn	20	ML

OR operator allowed

Result

ld	Name	Age	Race
DN001	John Chia	21	CN
DN002	Tom	18	CN
DN003	Jane	24	IND

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved. 13







 SELECT * FROM Customers WHERE Age BETWEEN 21 AND 24

The database table "Customers"

ld	Name	Age	Race
DN001	John Chia	21	CN
DN002	Tom	18	CN
DN003	Jane	24	IND
DN004	Shawn	20	ML

BETWEEN allows specification of a range of data

ld	Name	Age	Race
DN001	John Chia	21	CN
DN003	Jane	24	IND





 SELECT * FROM Customers WHERE Name LIKE 'Chia%'

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Chia Shoo	CN
DN003	Chia Sun	IND
DN004	Shawn	ML

% acts as a wildcard, to be used with LIKE

Result

ld	Name	Race
DN002	Chia Shoo	CN
DN003	Chia Sun	IND

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

15





 SELECT * FROM Customers WHERE Name LIKE '%Chia%'

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Chia Shoo	CN
DN003	Chia Sun	IND
DN004	Shawn	ML

ld	Name	Race
DN001	John Chia	CN
DN002	Chia Shoo	CN
DN003	Chia Sun	IND





 SELECT * FROM Customers WHERE Race IN ('ML','IND')

The database table "Customers"

ld	Name	Race
DN001	John Chia	CN
DN002	Tom	CN
DN003	Jane	IND
DN004	Shawn	ML

IN specifies a set of valid values

Result

ld	Name	Race
DN003	Jane	IND
DN004	Shawn	ML

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

17







 SELECT * FROM Customers WHERE CustId IN (SELECT CustId FROM Orders)

The database table "Customers"

Custld	Phone No
C1	4831
C2	4832
C3	4833



The database table "Orders"

Orderld	Custld	Order Date
T01	C1	1 May 06
T02	C1	2 May 06
T03	C2	14 May 06
T04	C4	16 May 06
T05	C5	18 May 06

Result

Custld	Phone No
C1	4831
C2	4832

Sub query is executed first, returning a set of row(s)





SELECT * FROM Customers WHERE Email IS NULL

The database table "Customers"

Id	Name	Email
DN001	John Chia	NULL
DN002	Chia Shoo	Shoo@gmail.com
DN003	Chia Sun	NULL
DN004	Shawn	shawn@gmail.com

IS NULL – retrieving rows whereby the value of the Column is null

Result

ld	Name	Email
DN001	John Chia	NULL
DN003	Chia Sun	NULL

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

19





SELECT * FROM Customers ORDER BY Age

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Order By specifies rows retrieved are to be sorted

ld	Name	Age
DN002	Tom	18
DN001	John Chia	21
DN003	Jane	24





SELECT * FROM Customers ORDER BY Age DESC

The database table "Customers"

Id	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Displayed in descending order

Result

ld	Name	Age
DN003	Jane	24
DN001	John Chia	21
DN002	Tom	18

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

21







SELECT Count(*) AS Total FROM Customers

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Count function returns the number of rows that satisfy the query command

Result Total 3



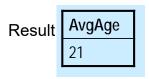


• SELECT AVG(Age) AS AvgAge FROM Customers

The database table "Customers"

ld	Name	Age
DN001	John Chia	21
DN002	Tom	18
DN003	Jane	24

Avg function returns the average of the values found in all the rows of the specified column



22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

23

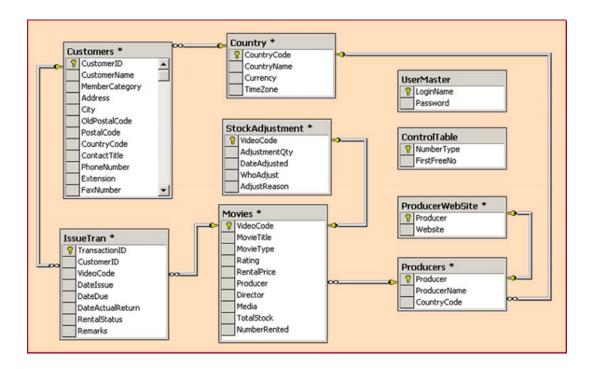




- Introduction
- Basic Queries
- Basic Queries (Illustrations with Dafesty Video Rental Shop)
- Advanced Queries
- Advanced Queries (Illustrations with Dafesty Video Rental Shop)







22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

25





- Movies Table
 - Stores all details pertaining to Movies like MovieTitle, MovieType, Rating, etc
- Customers Table
 - Stores all details pertaining to Customers like CustomerName, Address, etc
- IssueTran Table
 - Is a transaction table to record Video loan activities. A new record is created whenever a loan issue takes place. The same record is updated to a different status when the borrowed Video is returned.
 - Contains details like VideoCode, CustomerID, Datelssue, RentalStatus (in / out), ReturnDate, etc.
- Country Table
 - Stores a list of countries.
 - This is used for reference purposes for customer address.





- StockAdjustment Table
 - This table is provided to capture adjustments in stock usually performed when annual stock verification takes place.
- Producers Table
 - Stores details regarding Movie Producers like ProducerName, CountryCode,etc.
- ProducerWebSite Table
 - Stores the website details for Movie Producers.
 - Only those Movie Producers who have websites are included in this table.
- ControlTable Table
 - Stores the first free number for generating serial numbers where required.
 - For instance, a new Transaction ID in the IssueTran Table may be created based on the first free number value that is kept in this table.
- User Table
 - Stores the **Application** Users' name and password.
 - This may be used for application level authentication to use the Video Rental System written in ESNET module.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

27





- In addition to the tables shown on the ERD, the following tables are provided for exercises:
 - Documentaries Table
 - · Stores Video details (almost identical to Movies Table)
 - This table is used for Non-Movie Video Tapes
 - Employees Table
 - Stores details of all Employees in Dafesty Video Rental Private Ltd.
 - Details include name, age, salary, etc.
 - SalaryHistory Table
 - · Stores historical salary details of each employee.
 - A new record is created capturing the previous salary whenever a salary adjustment is made to an employee.





Select all rows and columns from a Table

SELECT * FROM Movies

- This statement retrieves all data from movies Table
 - SELECT indicates this is a data retrieval operation
 - The * stands for "all columns"
 - Movies is the name of the Table where the data resides
 - All rows in the Table will be retrieved since there are no conditions stipulated

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

29





Select all rows and columns from a Table

SELECT * FROM Movies

Output:

	VideoCode	MovieTitle	MovieType	Rating	Rental Pri
1	1	Star Trek 3: Search f	Sci-fi	PG	1.5
2	2	Star Trek 4: The Voya	Sci-fi	PG	1.5
3	3	Star Trek 5: The Fina	Sci-fi	PG	1.5
4	4	Demolition Man	Action	R	1.5
5	5	Nemesis	Action	R	1.5
6	6	Full Eclipse	Action	R	1.5
7,	7	Marked for Death	Action	U	1.5
1					F





Project: Select a vertical subset of the Table

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies

- This statement retrieves five columns from the movies Table
 - This statement demonstrates the retrieval of selected columns (not all columns) from a given Table
 - The columns are VideoCode, MovieTitle, MovieType, Rating, TotalStock
 - The sequence in which the columns are displayed is the same as specified in the column list.
 - All rows in the Table is retrieved since there are no conditions stipulated

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

31





Project: Select a vertical subset of the Table

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies

Output:

Five Columns Displayed. Sequence of column matches sequence listed in Select Statement

	VideoCode	MovieTitle	MovieType	Rating	TotalStock	_
1	1	Star Trek 3: Search f	Sci-fi	PG	7	
2	2	Star Trek 4: The Voya	Sci-fi	PG	0	
3	3	Star Trek 5: The Fina	Sci-fi	PG	3	
4	4	Demolition Man	Action	R	3	
5	5	Nemesis	Action	R	4	
6	6	Full Eclipse	Action	R	6	
7	7	Marked for Death	Action	U	3	▾





Project: Select a DISTINCT vertical subset of the Table

SELECT DISTINCT Rating
FROM Movies

- This statement retrieves a column from movies
 Table
 - This statement demonstrates the retrieval of selected column(s) from a given Table
 - DISTINCT requests that the data displayed have no duplicated row(s)

22/02/2018 1:24 pm - 03-sql select.pptx

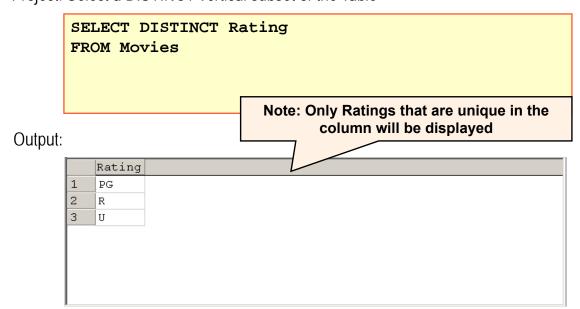
© 2018 National University of Singapore. All Rights Reserved.

33





Project: Select a DISTINCT vertical subset of the Table







Selection: Select a horizontal subset from the Table

```
SELECT * FROM Movies
WHERE Rating = 'PG'
```

- This statement retrieves specific rows from movies Table
 - Unlike earlier commands, where we obtained all rows, we now want to display only certain rows.
 - A condition has to be specified in the WHERE clause
 - Only rows that satisfy the condition is retrieved
 - In our database, only 126 out of 312 Movies satisfy the condition.

22/02/2018 1:24 pm - 03-sql select.pptx

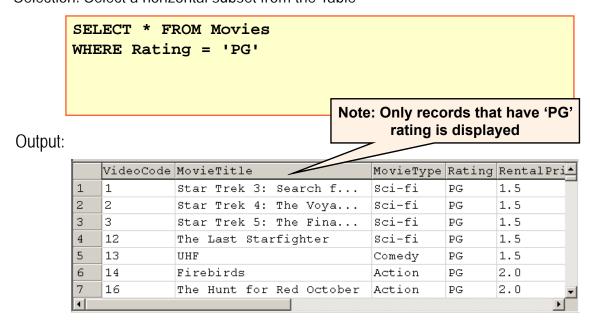
© 2018 National University of Singapore. All Rights Reserved.

35





Selection: Select a horizontal subset from the Table







Selection + Projection: Select a vertical and horizontal subset from the Table

SELECT VideoCode,MovieTitle,MovieType,Rating,TotalStock
FROM Movies
WHERE Rating = 'PG'

- This statement retrieves the specified columns and rows from Movies Table
 - This is a combination of earlier two queries that help us to obtain a vertical as well as horizontal subset.
 - The Table retrieved consists of 5 columns that satisfy the conditions in the WHERE clause

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

37

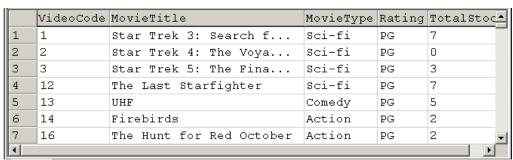




Selection + Projection: Select a vertical and horizontal subset from the Table

SELECT VideoCode,MovieTitle,MovieType,Rating,TotalStock
FROM Movies
WHERE Rating = 'PG'

Output:







NOT Condition

```
SELECT VideoCode,MovieTitle,MovieType,Rating,TotalStock
FROM Movies
WHERE NOT(Rating = 'PG')
```

- This statement retrieves the specified columns and rows from Movies Table
 - Like the previous example, the Table retrieved consists of 5 columns that satisfy the conditions in the WHERE clause
 - The NOT clause modifies the condition used in the SQL statement.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

39





NOT Condition

SELECT VideoCode,MovieTitle,MovieType,Rating,TotalStock
FROM Movies
WHERE NOT(Rating = 'PG')

Output:

The Records retrieved by this SQL query is mutually exclusive from the records retrieved by the previous SQL query. This SQL Query retrieve 186 records while the previous SQL query retrieved 126 records. Together, they are equivalent to the total number of records in the Movies Table

	VideoCode	MovieTitle	MovieType	Rating	TotalStock	_
1	4	Demolition Man	Action	R	3	
2	5	Nemesis	Action	R	4	
3	6	Full Eclipse	Action	R	6	
4	7	Marked for Death	Action	U	3	
5	8	Black Rain	Drama	R	4	
6	9	Red Heat	Action	R	4	
7	10	Die Hard	Action	R	2	▼





Using Alias: AS

SELECT MovieTitle AS Movie_Titles
FROM Movies

- This statement retrieves a column from movies Table
 - This statement demonstrates the retrieval of selected column(s) from a given Table
 - DISTINCT requests that the data displayed have no duplicated row(s)
 - AS redefines the name of the result set column.(ie Movie Title). It also allows the assignment of names to result set columns that have no names. (eg. Derived results set columns have no names)

22/02/2018 1:24 pm - 03-sql select.pptx

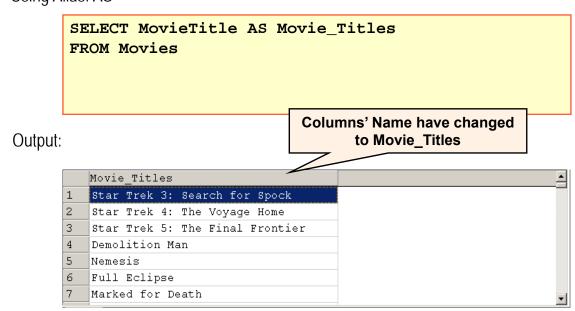
© 2018 National University of Singapore. All Rights Reserved.

41





Using Alias: AS







AND condition

```
SELECT * FROM Movies
WHERE Rating = 'PG'
AND MovieType = 'Sci-fi'
```

- This statement retrieves all data from Movies Table that satisfy the conditions in the WHERE clause
 - AND allows more conditions to be placed in the SELECT statement thereby limiting the total number of rows returned

22/02/2018 1:24 pm - 03-sql select.pptx

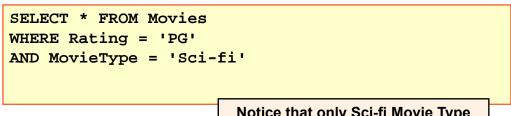
© 2018 National University of Singapore. All Rights Reserved.

43





AND condition



Output:

Notice that only Sci-fi Movie Type with 'PG' rating is selected

	VideoCode	MovieTitle	MovieType	Rating	Rental Pri
1	1	Star Trek 3: Search f	Sci-fi	PG	1.5
2	2	Star Trek 4: The Voya	Sci-fi	PG	1.5
3	3	Star Trek 5: The Fina	Sci-fi	PG	1.5
4	12	The Last Starfighter	Sci-fi	PG	1.5
5	88	Dune	Sci-fi	PG	2.0
6	97	My Science Project	Sci-fi	PG	2.0
7	117	Star Trek 6: The Undi	Sci-fi	PG	2.0
1					Þ





OR condition

```
SELECT * FROM Movies
WHERE Rating = 'U'
OR MovieType = 'Sci-fi'
```

- This statement retrieves all data from Movies Table that satisfy the conditions in the WHERE clause
 - OR allows either one of the conditions placed before and after it to be satisfied for any row in the Table to be retrieved

22/02/2018 1:24 pm - 03-sql select.pptx

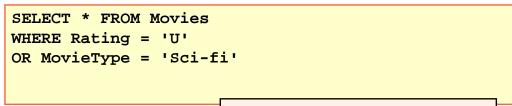
© 2018 National University of Singapore. All Rights Reserved.

45





OR condition



Output:

Notice only Sci-fi Movies or Movies with U Rating are displayed

				_		
	VideoCode	MovieTitle	MovieType	Rating	RentalPrice	Produc_
1	1	Star Trek 3: Search f	Sci-fi	PG	1.5	Fu Sho
2	2	Star Trek 4: The Voya	Sci-fi	PG	1.5	Kelvin
3	3	Star Trek 5: The Fina	Sci-fi	PG	1.5	Neo Ke
4	12	The Last Starfighter	Sci-fi	PG	1.5	Abdul
5	51	Blood Ties	Drama	U	2.0	Consta
6	54	Freejack	Sci-fi	PG	2.0	Ang Ki
7	88	Dune	Sci-fi	PG	2.0	Koh Ti
1						Þ





BETWEEN condition

SELECT * FROM IssueTran WHERE DateIssue BETWEEN '20 Nov 2000' AND '23 Nov 2000'

- This statement retrieves all data from IssueTran Table that satisfy the conditions in the WHERE clause
 - The BETWEEN keyword allows conditions, in the WHERE clause, to be defined as a range
 - SQL Server flexibly accepts different date formats
 - Refer to Books online for other formats

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

47





BETWEEN condition

SELECT * FROM IssueTran
WHERE DateIssue BETWEEN '20 Nov 2000' AND
'23 Nov 2000'

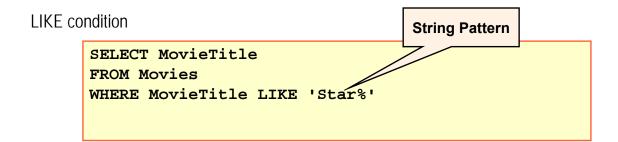
Output:

Note only 199 out of 400 rows are Retrieved

	TransactionID	CustomerID	Vid	DateIssue		DateDue_
1	2	4312	55	2000-11-20	00:00:00	2000-11
2	3	6598	94	2000-11-20	00:00:00	2000-11
3	4	1111	164	2000-11-20	00:00:00	2000-11
4	5	4312	291	2000-11-20	00:00:00	2000-11
5	6	8756	1	2000-11-20	00:00:00	2000-11
6	7	7856	254	2000-11-20	00:00:00	2000-11
7	8	8756	232	2000-11-20	00:00:00	2000-11
Ω 4	l a	1000	200	2000_11_20	00.00.00	2000_11







- This statement retrieves all data from Movies Table that satisfy the conditions in the WHERE clause
 - LIKE allow rows having values in the specified columns that matches the string pattern (ie Star%) to be retrieved
 - The "%" acts as a wildcard for other character(s) in the pattern. For example the wild card may be "s" (ie. Stars) or "dom" (ie Stardom)

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

49

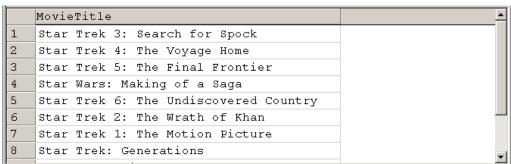




LIKE condition

SELECT MovieTitle
FROM Movies
WHERE MovieTitle LIKE 'Star%'

Output:







LIKE condition

SELECT MovieTitle FROM Movies
WHERE MovieTitle LIKE \%Star%'

- This statement retrieves all data from Movies Table that satisfy the conditions in the WHERE clause
 - The wildcard (%) placed before and after the string pattern (ie Star) allows values in the specified columns (ie MovieTitle column) of the Movies Table to be retrieved as long as the column string has the character(s) having the same string pattern (ie. Star)
 - The string pattern is not Cap Sensitive

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

51

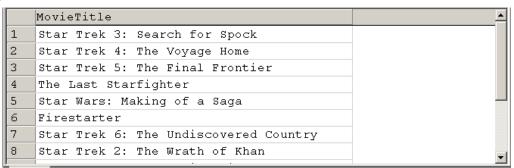




LIKE condition (with different sets of % sign)

SELECT MovieTitle FROM Movies WHERE MovieTitle LIKE '%Star%'

Output:







IN condition

```
SELECT * FROM Movies
WHERE Rating IN ('U','R')
```

- This statement retrieves all data from Movies Table that satisfy the conditions in the WHERE clause
 - The IN clause specifies a list of character(s) that values in the specified column (Rating column) should match.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

53





IN condition

```
SELECT *
FROM Movies
WHERE Rating IN ('U','R')
```

Output:

	VideoCode	MovieTitle	MovieType	Rating	Rental Pr_
1	4	Demolition Man	Action	R	1.5
2	5	Nemesis	Action	R	1.5
3	6	Full Eclipse	Action	R	1.5
4	7	Marked for Death	Action	U	1.5
5	8	Black Rain	Drama	R	1.5
6	9	Red Heat	Action	R	1.5
7	10	Die Hard	Action	R	1.5
۵ •	111	nia Bard 2	notion	ъ	1 5





Using Sub Queries

SELECT * FROM Producers
WHERE Producer IN(SELECT Producer
FROM ProducerWebSite)

- This statement retrieves all data from Producers Table that satisfy the conditions in the WHERE clause
 - The IN clause specifies a list of values, generated by another SELECT statement, that values in the specified column (ie Producer column from Producers Table) should match.
 - The SELECT statement in brackets are called sub query.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

55





Using Sub Queries

SELECT * FROM Producers
WHERE Producer IN(SELECT Producer
FROM ProducerWebSite)

Output:

	roducer	ProducerName	CountryCode	
1 2	20th	20th Century Fox Prod	UK	
2 C	Columbia	Columbia Pictures Pro	UK	
3 U	Jniversal	Universal Studio Prod	UK	





IS NULL condition

SELECT CustomerID, CustomerName, EmailAddress FROM Customers
WHERE EmailAddress IS NULL

- This statement retrieves three columns from Customers Table that satisfy the conditions in the WHERE clause
 - IS NULL specifies that the rows in the Table whose value in the specified column's must be a null in order for that row to be retrieved

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

57





IS NULL condition

SELECT CustomerID, CustomerName, EmailAddress FROM Customers
WHERE EmailAddress IS NULL

Output:

	CustomerID	CustomerName	EmailAddress
1	2131	Jon	NULL
2	2323	Richard Kwan	NULL
3	2345	Ng Teck Kie Anthony	NULL
4	2626	Steven Teo	NULL
5	5156	Lee Boon Kiat	NULL
6	6969	jason young	NULL
7	7856	Rajaram Venkatesh	NULL
8	8888	Kelvin Koh	NULL





ORDER BY clause

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies ORDER BY MovieTitle

- This statement retrieves five columns from movies Table that satisfy the conditions in the WHERE clause
 - ORDER BY specifies that rows retrieved are to be sorted in ascending order in the specified column (ie. MovieTitle)
 - ORDER BY, by default, sorts rows in ascending order. To sort by descending order, use ".... ORDER BY MovieTitle DESC"
 - ORDER BY allows sorting to be done in two or more columns. However, the first column that appears in the statement will have priority over the others. Eg. "ORDER BY MovieTitle, VideoCode"

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

59







ORDER BY clause

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies ORDER BY MovieTitle

Output:

Note: Sorted in Ascending order of Movie Title

	VideoCode	MovieTitle	MovieType	Rating	TotalStoc_
1	154	A Few Good Men	Drama	R	1
2	141	A League of Their Own	Drama	PG	5
3	49	Above the Law	Action	R	4
4	224	Absolute Power	Drama	R	1
5	115	Addams Family	Comedy	PG	4
6	27	Air America	Comedy	R	4
7	189	Airheads	Comedy	R	6
۵ • ا	228	Mladdin	Animation	тт	3





ORDER BY clause (Descending order)

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies ORDER BY MovieTitle DESC

- This statement retrieves five columns from movies
 Table that satisfy the conditions in the WHERE clause
 - ORDER BY specifies that rows retrieved are to be sorted in descending order in the specified column (ie. MovieTitle)
 - ORDER BY allows sorting to be done in two or more columns. However, the first column that appears in the statement will have priority over the others. Eg. "ORDER BY MovieTitle, VideoCode"

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

61





ORDER BY clause (Descending order)

SELECT VideoCode, MovieTitle, MovieType, Rating, TotalStock FROM Movies ORDER BY MovieTitle DESC

Output:

Note: Sorted in Descending order of Movie Title

	VideoCode	MovieTit	le	7	MovieType	Rating	TotalStock	_
1	282	X-Files	(Wetv	vired)	Drama	PG	4	
2	275	X-Files	(War	of the C	Drama	PG	4	
3	264	X-Files	(Toon	ns)	Drama	PG	6	
4	258	X-Files	(The	Host)	Drama	PG	3	
5	260	X-Files	(The	Erlenmey	Drama	PG	1	
6	272	X-Files	(The	Blessing Way	Drama	PG	6	
7	283	X-Files	(Tali	tha Cumi)	Drama	PG	6	•





COUNT function

SELECT COUNT(*) AS TotalNoOfMovies FROM Movies

- This statement retrieves all data from Movies Table
 - The COUNT function returns the "number of rows" that satisfy the query command: "SELECT * FROM Movies"
 - Using alias allows the naming of the columns that are derived.
 - COUNT and other functions are usually used in conjunction with GROUP BY clauses.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

63

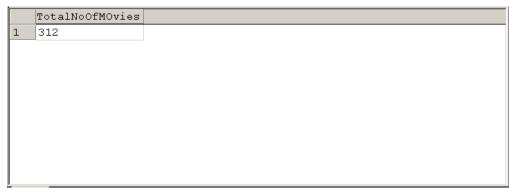




COUNT function

SELECT COUNT(*) AS TotalNoOfMovies FROM Movies

Output:







COUNT function

SELECT COUNT(Director) AS DirectorsInMovies FROM Movies

- This statement retrieves all data from Movies Table
 - The COUNT function returns the "number of rows" that satisfy the query command: "SELECT * FROM Movies"
 - Using alias allows the naming of the columns that are derived.
 - COUNT and other functions are usually used in conjunction with GROUP BY clauses.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

65





COUNT function

SELECT COUNT(Director) AS DirectorsInMovies FROM Movies

Output:

DirectorsInMovies
1 311





SUM function

SELECT SUM(TotalStock) FROM Movies

- This statement retrieves one columns from Movies Table
 - This column returns the sum of the values found in all the rows of the specified column (ie TotalStock column) of the Movies Table that satisfy the query: "SELECT * FROM Movies"
 - The specified column (ie TotalStock column) must have a data type that allows arithmetic operations
 - SUM and other functions are usually used in conjunction with GROUP BY clauses.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

67





SUM function

SELECT SUM(TotalStock) FROM Movies

Output:

(No column name)
1 1098





AVG function

SELECT AVG(RentalPrice) FROM Movies

- This statement retrieves one columns from Movies Table
 - This column returns the Average of the values found in all the rows of the specified column (ie RentalPrice column) of the Movies Table that satisfy the query: ""SELECT * FROM Movies"
 - The specified column (ie RentalPrice column) must have a data type that allows arithmetic operations
 - AVG and other functions are usually used in conjunction with GROUP BY clauses.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

69

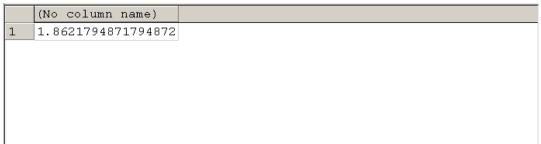




AVG function

SELECT AVG(RentalPrice) FROM Movies

Output:







- Introduction
- Basic Queries
- Basic Queries (Illustrations with Dafesty Video Rental Shop)
- Advanced Queries
- Advanced Queries (Illustrations with Dafesty Video Rental Shop)

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

71







 How do we retrieve name, id of the customers and video code they have rented?

table "Customers"

ld	Name
100	Chia
101	Derek
102	Esther
103	Venkat

table "IssueTrans"

Transld	CustomerId	VideoCode
T01	100	V01
T02	100	V02
T03	101	V11

Need to access values from two tables





 SELECT C.Id, C.Name, T.VideoCode FROM Customers C, IssueTrans T WHERE C.Id = T.CustomerId

table "Customers"

	ld	Name	
	100	Chia	
/	101	Derek	
	102	Esther	
	103	Venkat	

ld	Name	VideoCode
100	Chia	V01
100	Chia	V02
101	Derek	V11

Result

table "IssueTrans"

Transld	CustomerId	VideoCode
T01 (100	V01
T02 (100	V02
T03	101	V11

Inner join retrieve rows whose values exists in both tables

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

73





 How do I list details of <u>all</u> customers, and include video code for those who have made orders?

SELECT C.Id, C.Name, T.VideoCode
FROM Customers C LEFT OUTER JOIN IssueTrans T
ON C.Id = T.Customerld

Left table

Right table

Left Outer Join retrieved all rows from the left table and insert values from the right table (insert null if not found in the right table).







 SELECT C.Id, C.Name, T.VideoCode FROM Customers C LEFT OUTER JOIN IssueTrans T ON C.Id = T.CustomerId

table "Customers"

Id	Name
100	Chia
101	Derek
102	Esther
103	Venkat

table "IssueTrans"

Transld	CustomerId	VideoCode
T01	100	V01
T02	100	V02
T03	101	V11

ld	Name	VideoCode
100	Chia	V01
100	Chia	V02
101	Derek	V11
102	Esther	null
103	Venkat	null

Result

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

75







 How do I list the employees name with their supervisor name?

The database table "Employees"

ld	Name	ReportsTo
01	Lim	Null
02	MK Leong	01
03	Venkat	02
04	Derek	02







 SELECT staff.name, supervisor.name FROM Employees staff, Employees supervisor WHERE staff.reportsto = supervisor.id

The database table "Employees"

Id	Name	ReportsTo
01	Lim	Null
02	MK Leong	01
03	Venkat	02
04	Derek	02
05	Esther	02

self join: the table joins with itself

Result

Name	Name	
MK Leong	Lim	
Venkat	MK Leong	
Derek	MK Leong	
Esther	MK Leong	

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved







 SELECT staff.name, supervisor.name FROM Employees staff, Employees supervisor WHERE staff.reportsto = supervisor.id

The database table "Employees"

ld	Name	ReportsTo
01	Lim	Null
02	MK Leong	01
03	Venkat	02
04	Derek	02
05	Esther	02

staff

The staff "Lim" is not selected under staff column. How to reconstruct the query to include "Lim" in the staff list?

Result

supervisor

Name	Name
MK Leong	Lim
Venkat	MK Leong
Derek	MK Leong
Esther	MK Leong

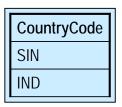




 SELECT CountryCode FROM Customers GROUP BY CountryCode
The database table "Customers"

Name	CountryCode	Category
Chia	SIN	А
Derek	SIN	В
Esther	SIN	В
Venkat	IND	А

Result



list the different country codes

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved. 79

뵭 Group By with Count





 SELECT CountryCode, Count(*) AS Num FROM Customers GROUP BY CountryCode

The database table "Customers"

Name	CountryCode	Category
Chia	SIN	А
Derek	SIN	В
Esther	SIN	В
Venkat	IND	А

List the country codes with the number of customers

Result

CountryCode	Num
SIN	3
IND	1





🛶 Group By and Having

 SELECT CountryCode, Count(*) AS Num FROM Customers GROUP BY CountryCode HAVING COUNT(*) > 1 The database table "Customers"

NameCountryCodeCategoryChiaSINADerekSINBEstherSINBVenkatINDA

list the country codes whereby there are more than 1 customers from the country

Result

CountryCode	Num
SIN	3

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

81







 SELECT CountryCode, Category FROM Customers GROUP BY CountryCode

The database table "Customers"

Name	CountryCode	Category
Chia	SIN	А
Derek	SIN	В
Esther	SIN	В
Venkat	IND	А

Compilation Error (SQL Server)! Category is not in the Group-By list







🖶 Group By and Count

• How do we retrieve the country code with more customers than "IND"?

The database table "Customers"

Name	CountryCode	Category
Chia	SIN	А
Derek	SIN	В
Esther	SIN	В
Venkat	IND	А

SELECT CountryCode FROM Customers **GROUP BY CountryCode** HAVING Count(*) > (SELECT Count(*) FROM Customers WHERE CountryCode = 'IND')

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved. 83



Join And Group By





 SELECT C.Id, Count(*) As Num FROM Customers C, IssueTrans T WHERE C.Id = T.CustomerId GROUP BY C.Id

_	table Customers	
	ld	Name
	100	Chia
	101	Derek
	102	Esther
	103	Venkat

Id	Num
100	2
101	1

Result

table '	"Issue"	Trans"

Transld	CustomerId	VideoCode
T01 (100	V01
T02 🤇	100	V02
T03	101	V11





SELECT * FROM USA-Customers UNION SELECT * FROM Europe-Customers

The database table "USA-Customers"

Custld	Phone No
U1	4831
U2	4832

The database table "Europe-Customers"

Custld	Phone No
E1	305656
E2	456677

Result

Phone No	
U1 4831	
4832	
305656	
456677	

Union return a merger of multiple tables columns with compatible data type

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved. 85







SELECT TOP 2 CustomerID FROM Customers

The database table "Customers"

Custld	Phone No
U1	4831
U2	4832
U3	4678
U4	4568

TOP *n* returns the first nth rows in the results

Result







- Introduction
- Basic Queries
- Basic Queries (Illustrations with Dafesty Video Rental Shop)
- Advanced Queries
- Advanced Queries (Illustrations with Dafesty Video Rental Shop)

© 2018 National University of Singapore. All Rights Reserved.

87





GROUP BY clause

SELECT MovieType FROM Movies GROUP BY MovieType

- This statement retrieves a column from Movies Table
 - GROUP BY aggregates the returned rows that have the same value in the column (MovieType column) specified in the GROUP BY clause
 - GROUP BY is constantly used with functions such as COUNT, SUM and AVG





GROUP BY clause

SELECT MovieType
FROM Movies
GROUP BY MovieType

Output:



Records Affected: 33

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

89





GROUP BY clause with COUNT function

SELECT COUNT(MovieType) AS NumberOfMovies, MovieType FROM Movies GROUP BY MovieType

- This statement retrieves a column from Movies table
 - GROUP BY aggregates the returned rows that have the same value in the column (MovieType column) specified in the GROUP BY clause
 - COUNT(MovieType) in this case returns the "number of rows" aggregated for each returned value in MovieType column





GROUP BY clause with COUNT function

SELECT COUNT(MovieType) AS NumberOfMovies, MovieType FROM Movies GROUP BY MovieType

Output:

	NumberOfMovies	MovieType
1	83	Action
2	10	Adventure
3	4	Animated
4	8	Animation
5	51	Comedy
6	1	Documentar
7	121	Drama
8	4	Horror

Records Affected: 33

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

91





HAVING clause

SELECT COUNT(MovieType) AS NumberOfMovies, MovieType FROM Movies GROUP BY MovieType HAVING COUNT(MovieType) > 50

- This statement retrieves a column from Movies Table
 - COUNT(...) aggregates the returned rows based on each row having the same values as other rows with respect to a specified column(s) (MovieType column)
 - HAVING specifies a search condition that will be factored in the rows returned after the GROUP BY clause has taken effect





HAVING clause

SELECT COUNT(MovieType) AS NumberOfMovies, MovieType
FROM Movies
GROUP BY MovieType
HAVING COUNT(MovieType) > 50

Output:

	NumberOfMovies	MovieType
1	83	Action
2	51	Comedy
3	121	Drama
1		
1		
1		

Records Affected: 3

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

93





SUM function with GROUP BY

SELECT SUM(TotalStock) AS TotalNumberOfMovies, Rating FROM Movies
GROUP BY Rating

- This statement retrieves two columns from Movies Table
 - The two columns are TotalStock and Rating
 - The SUM function adds up all the values in the TotalStock column where the same Rating are grouped together
 - TotalStock column must have a data type that allows arithmetic operations





SUM function with GROUP BY

SELECT SUM(TotalStock) AS TotalNumberOfMovies, Rating FROM Movies
GROUP BY Rating

Output:

	TotalNumberOfMovies	Ratino
1	486	PG
2	549	R
3	63	U
3	_ 63	U
1		
1		
1		

Records Affected: 3

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

95





AVG function with GROUP BY

SELECT AVG(RentalPrice) AS AveragePriceOfMovie,Rating FROM Movies
GROUP BY Rating

- This statement retrieves two columns from Movies Table
 - · The two columns are RentalPrice and Rating
 - The AVG function adds up all the values in the RentalPrice column, where the same Rating are grouped together, and then divides them by the number of rows used for each Rating
 - TotalStock column must have a data type that allows arithmetic operations





AVG function with GROUP BY

SELECT AVG(RentalPrice) AS AveragePriceOfMovie,Rating FROM Movies
GROUP BY Rating

Output:

		AveragePriceOfMovie	Rating
	1	1.8928571428571428	PG
	2	1.8353293413173652	R
ı	3	1.8947368421052631	U
ı		1.0747300421032031	0
ı			
ı			
1			
- 1			
1			
, J			

Records Affected: 33

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

97





Joining Tables: INNER JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM Producers
INNER JOIN ProducerWebSite
ON ProducerWebSite.Producer = Producers.Producer

--ALTERNATIVE SYNTAX OF INNER JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM Producers,ProducerWebSite
WHERE ProducerWebSite.Producer = Producers.Producer





- This statement retrieves three columns from ProducerWebSite and Producers Table.
 - When retrieving rows from two or more Tables that have the same column name, qualify (ie differentiate) the column to be retrieved by giving the Table name followed by the column name (Producers.Producer)
 - Inner join is an inclusive join. It retrieved rows whose value exists in *both* Tables. In other words, if either one of the Table have a rows whose values (ie Joining Condition) is not found in the other Table, that row will not be retrieved.
 - Syntactically, INNER JOIN can be represent with just JOIN because the default JOIN is the INNER JOIN.
 - The WHERE condition can be used in lieu of the INNER JOIN. Refer to the alternative syntax.
 - The WHERE condition is not a filter as in previous cases but acts as a condition for joining. It is often referred as JOIN condition.

© 2018 National University of Singapore. All Rights Reserved.

99





Joining Tables: INNER JOIN

SELECT Producers.Producer,ProducerName,WebSite FROM Producers

INNER JOIN ProducerWebSite

ON ProducerWebSite.Producer = Producers.Producer

Output:

	Producer	ProducerName	WebSite
1	20th	20th Century Fox Prod	www.century.com
2	Columbia	Columbia Pictures Pro	www.columbia.com
3	Universal	Universal Studio Prod	www.universal.com





Joining Tables: LEFT OUTER JOIN

SELECT Producers.Producer,ProducerName,WebSite

FROM Producers

LEFT OUTER JOIN ProducerWebSite

Left Table

ON Producers.Producer = ProducerWebSite.Producer

- This statement retrieves three columns from ProducerWebSite and Producers Table.
 - · OUTER JOIN consists of the LEFT and RIGHT outer joins.
 - The LEFT OUTER JOIN retrieved all the rows from the Table on the left side of the join clause (ie. Producers Table) and attempts to insert values from the Table (ie ProducerWebSite Table) on the right side of the clause.
 - OUTER JOIN is an exclusive join. In other words, if the column values based on the ioin condition cannot be found, NULL is displayed.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

101





Joining Tables: LEFT OUTER JOIN

SELECT Producers.Producer,ProducerName,WebSite FROM Producers

LEFT OUTER JOIN ProducerWebSite

ON Producers.Producer = ProducerWebSite.Producer

Output:

Producer	ProducerName	WebSite
20th	20th Century Fox Prod	www.century.com
Columbia	Columbia Pictures Pro	www.columbia.com
George	George Lucas Production	NULL
Pixar	Pixar Entertainment	NULL
Raintree	RainTree Pictures	NULL
Universal	Universal Studio Prod	www.universal.com
Walt	Walt Disney Studio	NULL
Warner	Warner Brothers Produ	NULL





Joining Tables: SELF JOIN

SELECT D1.VideoTitle AS DocumentaryTitle,
D2.VideoTitle AS DocumentaryPrequel

FROM Documentaries D1

INNER JOIN Documentaries D2

ON D2.VideoCode = D1.PreviousEpisode

--ALTERNATIVE SYNTAX

SELECT D1.VideoTitle AS DocumentaryTitle,
D2.VideoTitle AS DocumentaryPrequel

FROM Documentaries D1, Documentaries D2

WHERE D2.VideoCode = D1.PreviousEpisode

- This statement retrieves two columns from Documentaries Table.
 - A self join is required when some conditions require that the Table to join with itself
 - An Alias is required when performing SELF JOINS so that the same Table can be recognised as different Tables.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

103





Joining Tables: SELF JOIN

SELECT D1.VideoTitle AS DocumentaryTitle,
D2.VideoTitle AS DocumentaryPrequel

FROM Documentaries D1

INNER JOIN Documentaries D2

ON D2.VideoCode = D1.PreviousEpisode

Output:

	DocumentaryTitle	DocumentaryPrequel
1	Poverty in Russia 2	Poverty in Russia 1
2	Lonely Planet: Paris 2	Lonely Planet: Paris 1





Joining Columns: UNION

SELECT MovieTitle FROM Movies
UNION
SELECT VideoTitle FROM Documentaries

- This statement retrieves a column from Customers and Producers Table.
 - UNION allows two or more Table columns to be merged into one as long as the datatype for the columns are compatible.
 - The values in the columns are by default DISTINCT rather than ALL. In other words, the retrieved values in the specified columns will not be repeated unless stated otherwise

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

105

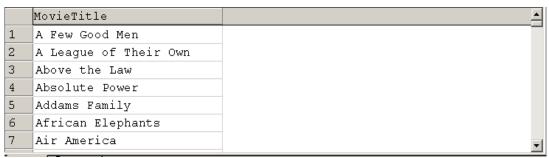




Joining Columns: UNION

SELECT MovieTitle FROM MovieS UNION SELECT VideoTitle FROM DocumentarieS

Output:







Limiting Result Set: TOP n

SELECT TOP 5 CustomerID, COUNT(CustomerID) AS NoOfTransMade FROM IssueTran
GROUP BY CustomerID
ORDER BY NoOfTransMade DESC

- This statement retrieves two columns from IssueTran Table
 - COUNT(...) sums the number of rows in the Specified Column (ie CustomerID Column) when the GROUP BY condition is placed on the SQL query.
 - ORDER BY arranges the results in Descending order.
 - Finally, TOP n (ie 5 in the above query) displays the first nth rows in the results of the SQL query.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

107





Limiting Result Set: TOP n

SELECT TOP 5 CustomerID, COUNT(CustomerID) AS NoOfTransMade
FROM IssueTran
GROUP BY CustomerID
ORDER BY NoOfTransMade DESC
Only Top 5 Rows of the Result Set is

Output:

CustomerID NoOfTransactionsMade

1 6542 14

2 8756 14

3 1003 13

4 2345 13

5 2270 11

Records Affected: 5

displayed.





- Sub query vs Joins
 - There are a lot of situations when joins can be used in lieu of sub queries to obtain the desired result set.
 - For example, in the situation where it is required to List the Movie Titles and Ratings of all Movies borrowed
 - Using a Sub query, we have

```
SELECT Movietitle FROM Movies, Issuetran
WHERE Movies.Videocode = Issuetran.Videocode
and CustomerID = 9999
```

Using a Join, we have

```
SELECT MovieTitle FROM Movies
WHERE videocode IN
(SELECT Videocode FROM Issuetran WHERE Customerid=9999 )
```

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

109

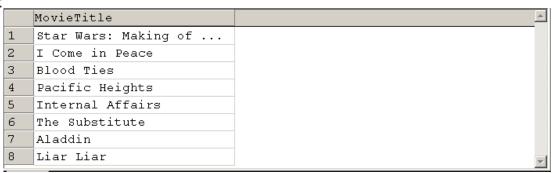




Either way, the result set returned is the same.

SELECT Movietitle FROM Movies, Issuetran WHERE Movies.Videocode = Issuetran.Videocode and CustomerID = 9999

Output:









- Sub query vs Joins
 - However, there are circumstances when a Join cannot be used in lieu of Sub queries.
 - One such circumstance is when an aggregation of rows is used as a condition for the query. For example:

```
SELECT MovieTitle,RentalPrice
FROM Movies
WHERE RentalPrice > (SELECT AVG(RentalPrice) FROM Movies)
```

 Note that the result set using the above query cannot be repeated using a query with Joins.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.





- Some considerations in deciding to use Sub query vs Joins:
 - Use Join when data is drawn from two tables
 - Use sub query when comparison is made to an aggregation of second table
 - When either sub query or joins can be used:
 - Choose the one that seems natural to an English statement or to you.
 - Sub query is simplier to develop since sub query can be independently tested
 - However, the speed of retrieval using joins (on index fields) could be faster





- Revision of Key Concepts
- Discussion on Workshop
- Questions and Answer
- Assignments and Home Work



© 2018 National University of Singapore. All Rights Reserved.





- Apart from the Left Outer Join, we can also use a Right Outer Join, Full Join and Cross Join on tables/views for retrieving records.
- We will be discussing these Joins in the following order:
 - Right Outer Join
 - Full Join
 - Cross Join





Joining Tables: RIGHT OUTER JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM ProducerWebSite
RIGHT OUTER JOIN Producers Right Table
ON Producers.Producer = ProducerWebSite.Producer

- This statement retrieves three columns from ProducerWebSite and Producers Table.
 - The RIGHT OUTER JOIN is similar to the LEFT OUTER JOIN except that the Table on the right of the join clause (Producers Table) will display all the rows in it irregardless of whether there is row with similar value in the other table.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

115





Joining Tables: RIGHT OUTER JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM ProducerWebSite
RIGHT OUTER JOIN Producers
ON Producers.Producer = ProducerWebSite.Producer

Output:

Notice that Producer records in Producers Table that are not found in the ProducersWebSite Table will be displayed as null in the Result set

	Producer	Produce	WebSite	
1	20th	20th Century Fo	www.century.com	
2	Columbia	Columbia Pictures Pro.	www.columbia.com	
3	George	George Lucas Production	NULL	
4	Pixar	Pixar Entertainment	NULL	
5	Raintree	RainTree Pictures	NULL	
6	Universal	Universal Studio Prod	www.universal.com	
7	Walt	Walt Disney Studio	NULL	
8	Warner	Warner Brothers Produ	NULL	





Joining Tables: FULL JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM ProducerWebSite
FULL JOIN Producers
ON Producers.Producer = ProducerWebSite.Producer

- This statement retrieves three columns from ProducerWebSite and Producers Table.
 - The FULL JOIN is similar to the equivalent of a LEFT OUTER JOIN followed by a RIGHT OUTER JOIN
 - There is no difference between the left and right table when a Full Join is applied to the tables to be joined.
 - There is no alternate syntax for a FULL JOIN.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

117





Joining Tables: FULL JOIN

SELECT Producers.Producer,ProducerName,WebSite
FROM ProducerWebSite
FULL JOIN Producers
ON Producers.Producer = ProducerWebSite.Producer

Output:

The results between a Full Join and a Right Outer Join is the same in this case because there is no Producer that is found in ProducerWebSite Table but not Producers Table.

	Producer	ProducerName	WebSite
1	20th	20th Century Fox Prod	www.century.com
2	Columbia	Columbia Pictures Pro	www.columbia.com
3	George	George Lucas Production	NULL
4	Pixar	Pixar Entertainment	NULL
5	Raintree	RainTree Pictures	NULL
6	Universal	Universal Studio Prod	www.universal.com
7	Walt	Walt Disney Studio	NULL
8	Warner	Warner Brothers Produ	NULL





Joining Tables: CROSS JOIN

SELECT P1.Producer, P2.WebSite FROM ProducerWebSite P1 CROSS JOIN ProducerWebSite P2

- This statement retrieves two columns from the ProducerWebSite Table.
 - A CROSS JOIN produces a Cartesian product (ie all possible combinations) of the records on both side of the JOIN.
 - In other words if there are three records from each table respectively, a Cartesian product of all records would return a results set having the multiple of three and three (ie. nine) records.
 - Note that there is no Joining Condition (ie ON clause) for the CROSS JOIN because we are showing all possible combinations in the result set.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

119





Joining Tables: CROSS JOIN

SELECT P1.Producer, P2.WebSite FROM ProducerWebSite P1 CROSS JOIN ProducerWebSite P2

Since there are three records in the ProducerWebSite Table, a Cartesian Product of same Employees Table would result in a result set with nine records.

Notice that the Producer "20th" and the website

"www.century.com" appears three times each.

Output:

		www.centul	y.com appears three times each.
	Producer	WebSite	
1	20th	www.century.com	
2	Columbia	www.century.com	
3	Universal	www.century.com	
4	20th	www.columbia.com	
5	Columbia	www.columbia.com	
6	Universal	www.columbia.com	
7	20th	www.universal.com	
-			





- Alternate syntax of a CROSS JOIN
 - You will be surprise that the syntax of a cross join is as follows:

```
SELECT P1.Producer, P2.WebSite
FROM ProducerWebSite P1
CROSS JOIN ProducerWebSite P2
```

 Note that there are no WHERE clause in the Query statement.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.





- If there are any field names with spaces, MS SQL Server have problems reading the field name.
 - Example of a field names with spaces are Order Details table in Northwind database.
 - In order to query the table, we need to place the field name in a pair of square brackets. (the square brackets acts as an escape sequence)
 - For example, we do not: SELECT * FROM Order Details
 - Instead we : SELECT * FROM [Order Details]





- This appendix serves to provide information regarding functions that are available in SQL Server.
- Functions allows users to simplify many of their operations.
 - For example users can use COUNT function to return the number of rows returned by the query.
- The Functions that will be discussed are:
 - Aggregate Functions:
 - Functions which operate on a set of records to return a single but summarizing value.
 - Scalar Functions:
 - Functions which operate on a single set of record and return a single value.
 - Some Scalar functions include Mathematical Functions, String Functions and System Functions.
 - A brief description for these functions are provided here.
 Refer to Help Online for more details.

© 2018 National University of Singapore. All Rights Reserved.





- Aggregate Functions:
 - Commonly used Aggregate functions are as follows:

Aggregate Function	Function Description
AVG	Returns the average of the values in a group. Null values are ignored.
COUNT	Returns the number of items in a group.
MAX	Returns the maximum value in the expression.
MIN	Returns the minimum value in the expression.
SUM	Returns the sum of all the values, or only the DISTINCT values, in the expression. SUM can be used with numeric columns only. Null values are ignored.







- Mathematical Functions:
 - Performs a calculation based on input values provided as parameters to the function, and returns a numeric value.
 - Commonly used Mathematical functions are as

Mathematical Function	Function Description
CEILING	Returns the smallest integer greater than, or equal to, the given numeric expression.
FLOOR	Returns the largest integer less than or equal to the given numeric expression.
RAND	Returns a random float value from 0 through 1.
ROUND	Returns a numeric expression, rounded to the specified length or precision.

© 2018 National University of Singapore. All Rights Reserved.







- Date and Time Functions:
 - These scalar functions perform an operation on a date and time input value and return a string, numeric, or date and time value.
 - · Commonly used Date and Time functions are as

Date and Time Function	Function Description
DATEADD	Returns a new datetime value based on adding an interval to the specified date.
DATEDIFF	Returns the number of date and time boundaries crossed between two specified dates.
GETDATE	Returns the current system date and time in the Microsoft® SQL Server™ standard internal format for datetime values.
DAY / MONTH / YEAR	Returns an integer that represents the day / month / year part of a specified date respectively







- String Functions:
 - These scalar functions perform an operation on a string input value and return a string or numeric value.
 - Commonly used String functions are as follows:

String Function	Function Description
ASCII	Returns the ASCII code value of the leftmost character of a character expression.
LEFT	Returns the part of a character string starting at a specified number of characters from the left.
LEN	Returns the number of characters, rather than the number of bytes, of the given string expression, excluding trailing blanks.
LOWER	Returns a character expression after converting uppercase character data to lowercase.

© 2018 National University of Singapore. All Rights Reserved.

127





• String Functions (continued):

String Function	Function Description
LTRIM	Returns a character expression after removing leading blanks.
RIGHT	Returns the part of a character string starting a specified number of <i>integer_expression</i> characters from the right.
RTRIM	Returns a character string after truncating all trailing blanks.
SUBSTRING	Returns part of a character, binary, text, or image expression.
STR	Returns character data converted from numeric data.
UPPER	Returns a character expression with lowercase character data converted to uppercase.







- System Functions:
 - These scalar functions perform operations on and return information about values, objects, and settings in Microsoft® SQL Server™.
 - Commonly used System functions are as follows:

System Function	Function Description
CAST & CONVERT	Explicitly converts an expression of one data type to another. CAST and CONVERT provide similar functionality.
ISDATE	Determines whether an input expression is a valid date.
ISNULL	Replaces NULL with the specified replacement value.
ISNUMERIC	Determines whether an expression is a valid numeric type.

© 2018 National University of Singapore. All Rights Reserved.





- System Functions (continued):
 - Cast and Convert are some of the common functions to handle different data types. (Example, from smalldatetime to String or double to integer.)
 - Changing data types allow operations that was previously impossible.
 - For example, smalldatetime type cannot be concatenated with a string datatype. Hence Casting is used to allow smalldatetime data type to be cast/converted to a string before it is concatenated with another string.







- System Functions (continued):
 - Example of using CAST:

```
SELECT DISTINCT DateIssue, CAST(Dateissue AS nvarchar(20)) AS [Casted DateIssue]
FROM Issuetran
WHERE CUSTOMERID = 9999
```

Output:

	DateIssue		Cast	ed	Date1	[ssue
1	2000-11-23	00:00:00	Nov	23	2000	12:00AM
2	2000-11-24	00:00:00	Nov	24	2000	12:00AM
3	2000-11-25	00:00:00	Nov	25	2000	12:00AM
4	2000-11-26	00:00:00	Nov	26	2000	12:00AM
4	2000-11-20	00.00.00	NOV	20	2000	12.00AM

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

131





- System Functions (continued):
 - Example of using CONVERT:

```
SELECT DISTINCT DateIssue,
CONVERT(nvarchar(20),Dateissue,101) AS [Converted DateIssue]
FROM Issuetran
WHERE CUSTOMERID = 9999
```

Output:

		DateIssue		Converted	DateIssue
1	-	2000-11-23	00:00:00	11/23/2000)
2	?	2000-11-24	00:00:00	11/24/2000)
3	}	2000-11-25	00:00:00	11/25/2000)
4	Į	2000-11-26	00:00:00	11/26/2000)
ľ		2000 11 20	00.00.00	11, 20, 2000	,
ı					
ı					





- Data types that are used by MS SQL Server:
 - A brief description for these data types are provided here. Refer to Help Online for more details.

Data Types	Data Type Description	
bigint	Integer (whole number) data from -2^63 (-9223372036854775808) through 2^63-1 (9223372036854775807).	
int	Integer (whole number) data from -2^31 (-2,147,483,648) through 2^31 - 1 (2,147,483,647).	
smallint	Integer data from 2^15 (-32,768) through 2^15 - 1 (32,767).	
tinyint	Integer data from 0 through 255.	
bit	Integer data with either a 1 or 0 value.	

© 2018 National University of Singapore. All Rights Reserved.

133





 Data types that are used by MS SQL Server (continued):

Data Types	Data Type Description
decimal	Fixed precision and scale numeric data from -10^38 +1 through 10^38 –1.
numeric	Functionally equivalent to decimal .
money	Monetary data values from -2^63 (-922,337,203,685,477.5808) through 2^63 - 1 (+922,337,203,685,477.5807), with accuracy to a tenthousandth of a monetary unit.
float	Floating precision number data from -1.79E + 308 through 1.79E + 308.
real	Floating precision number data from -3.40E + 38 through 3.40E + 38.







 Data types that are used by MS SQL Server(continued):

Data Types	Data Type Description
datetime	Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds.
smalldatetime	Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute.
char	Fixed-length non-Unicode character data with a maximum length of 8,000 characters.
varchar	Variable-length non-Unicode data with a maximum of 8,000 characters.
nchar	Fixed-length Unicode data with a maximum length of 4,000 characters.

22/02/2018 1:24 pm - 03-sql select.pptx

© 2018 National University of Singapore. All Rights Reserved.

135





 Data types that are used by MS SQL Server(continued):

Data Types	Data Type Description
nvarchar	Variable-length Unicode data with a maximum length of 4,000 characters. sysname is a system-supplied user-defined data type that is functionally equivalent to nvarchar(128) and is used to reference database object names.
binary	Fixed-length binary data with a maximum length of 8,000 bytes.
varbinary	Variable-length binary data with a maximum length of 8,000 bytes.
Image	Variable-length binary data with a maximum length of 2^31 - 1 (2,147,483,647) bytes.
timestamp	A database-wide unique number that gets updated every time a row gets updated.







© 2018 National University of Singapore. All Rights Reserved.